2012 IEEE Ninth International Conference on Services Computing

Stock Market Volatility Prediction: A Service-Oriented Multi-Kernel Learning Approach

Feng Wang
State Key Lab of Software Engineering
Wuhan University
Wuhan, China
Email: fengwang@whu.edu.cn

Ling Liu
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
Email: lingliu@cc.gatech.edu

Chenxiao Dou
School of Computer Science
Wuhan University
Wuhan, China
Email: 2008302530054@whu.edu.cn

***Abstract*—Stock market is an important and active part of nowadays financial markets. Stock time series volatility analysis is regarded as one of the most challenging time series forecasting due to the hard-to-predict volatility observed in worldwide stock markets. In this paper we argue that the stock market state is dynamic and invisible but it will be influenced by some visible stock market information. Existing research on financial time series analysis and stock market volatility prediction can be classified into two categories: in depth study of one market factor on the stock market volatility prediction or prediction by combining historical price fluctuations with either trading volume or news. In this paper we present a service-oriented multi-kernel based learning framework (MKL) for stock volatility analysis. Our MKL service framework promotes a two-tier learning architecture. In the top tier, we develop a suite of data preparation and data transformation techniques to provide a source-specific modeling, which transforms and normalizes a source specific input dataset into the MKL ready data representation. Then we apply data alignment techniques to prepare the datasets from multiple information sources based on the classification model we choose for cross-source correlation analysis. In the next tier, we develop model integration methods to perform three analytic tasks: (i) building one sub-kernel per source, (ii) learning and tuning the weights for sub-kernels through weight adjustment methods and (iii) performing multi-kernel based cross-correlation analysis of market volatility. To validate the effectiveness of our service oriented MKL approach, we performed experiments on HKEx 2001 stock market datasets with three important market information sources: historical prices, trading volumes and stock related news articles. Our experiments show that 1) multi-kernel learning method has a higher degree of accuracy and a lower degree of false prediction, compared to existing single kernel methods; and 2) integrating both news and trading volume data with historical stock price information can significantly improve the effectiveness of stock market volatility prediction, compared to many existing prediction methods.**

***Keywords*-multiple kernel learning; stock prediction; support vector machine; multi-data source integration;**

I. Introduction

Stock market is an important and active part of nowadays financial markets. Since finance market has become more and more competitive, stock price prediction has been a hot research topic in the past few decades. Autoregressive and moving average are some of the popular stock volatility prediction techniques, which have dominated for years in time series analysis. With the advancement of computing technology, data mining techniques have been widely used for the stock price prediction. Several approaches using inductive learning for prediction have been developed using historical stock price data, such as k-nearest neighbor and neural network, which have greatly improved the performance of prediction. However, one major weakness of these existing approaches is that they only rely on the historical price, and neglect some other information and their influences on the market volatility.

Many factors observed from other data sources tend to be less or not aligned well to the time-series information of the stock data such as historical prices, such as news about big mergers, bankruptcy of some companies or unexpected election and economic turmoil. However, such sources of information may have significant impacts on the time series behavior of the market. In recent years, several researchers have engaged in studying stock price prediction by combining historical prices with news to help improve the performance of prediction [4]. Most of them have applied or extended existing data mining techniques to explore how and when the market news may influence the investors' actions and in-turn affect the stock prices. At the same time, other researchers have argued that the trading volume is an important source of information, which may reflect some actions taken by the investors in stock trading. Some preliminary studies have been conducted to show the impact of stock volume fluctuation on the stock price movements. Some other studies have investigated the impact of combining the trading volume with historical prices on the effectiveness of stock volatility prediction. Surprisingly, none of the existing research, to the best of our knowledge, has explored whether correlations of fluctuations in historical prices, in trading volumes and in news (bad verse good news) may provide more and stronger indicators and improve accuracy of stock market volatility predication.

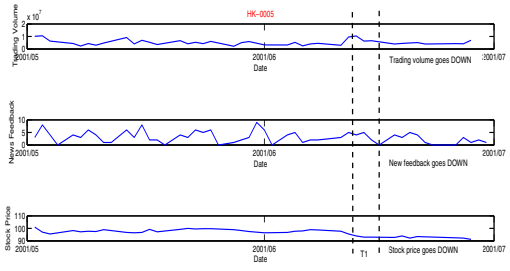
In this paper we argue that combining historical price fluctuations with two or more sources of information, such as both volume and news, can significantly improve the quality of stock market volatility prediction. For example, the correlation analysis of more than two sources of information will provide stronger indicators than correlating historical price with news independently from the correlation analysis of historical price with volume. This is because when historical price and news fail to show good correlation during a time interval, the predication based solely on price and news may fail, but if the correlation between historical price and volume during the same time interval is strong, then we can still deliver high quality predication with high confidence. Based on this

978-0-7695-4753-4/12 $26.00 © 2012 IEEE
DOI 10.1109/SCC.2012.35

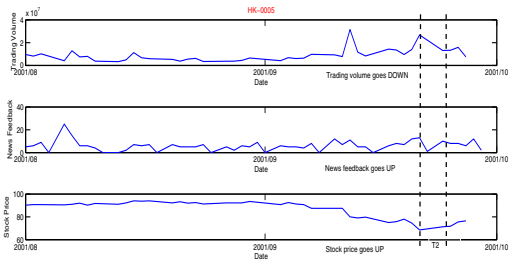
intuition and our observations over real stock trading datasets and related news datasets, we argue that the more sources of time series information related to stock market we use, the higher quality of the stock market volatility prediction may produce. Our experimental results show that using all three sources of time series data, historical prices, trading volume and news, one can significantly improve the stock market volatility predication over the predication based on one or two sources of information. Our second argument is the need for developing systematic framework for integrating the N ($N \geq 3$) factors by taking into account three types of inconsistencies: (i) the inconsistency observed within the same source of information, (ii) the inconsistency observed in the correlation of historical price with one other source, and (iii) the inconsistency observed when correlating all n sources of information.

For example, all existing work, which combines historical price fluctuations with news or historical price fluctuations with trading volume fluctuations for the market prediction have encountered some difficult dilemma in terms of prediction consistency. Figure 1(a) shows that the stock prices are strongly correlated with the feedback of news and trading volume during the time period $T1$. More interestingly, whether we choose to combine historical prices with news articles or with trading volumes is not critical due to that fact that both news and volume have strong correlations with the historical price fluctuations, the prediction result will be consistent with the actual price movement in the market. However, for some other datasets or some different periods of the same dataset, we may observe inconsistency or possibly conflicting results with respect to the positive or negative movements of news feedback, the high or low of trading volumes and how they relate to the ups and downs of stock prices. This is primarily due to the difference in terms of the correlation strength and the ways of how correlation intervals in different time series data sources are established. Figure 1(b) shows that for stock 0005.HK, using the news and historical price we can make a correct prediction during the time period T_2, while using trading volume will produce incorrect prediction during the same time period. In other cases, using news with historical price will generate incorrect prediction while combining transaction volume with historical price will provide better prediction accuracy [3]. Surprisingly, very few have engaged in investigating the problem of integrating more than two types of information sources and studying how to utilize multiple information sources to improve the accuracy and consistency in stock market volatility prediction.

In this paper we develop a service oriented multi-kernel learning method. We argue that it is not only necessary but also critical to investigate a general, multiple kernel based methodology that utilizes a weight function with weight update methods to incorporate and balance the contributions from multiple data sources in the market volatility prediction. We conjecture that to deliver multi-kernel learning based stock volatility analysis as a service, we need to address two major technical challenges: (1) How to extract the information



(a) Stock 0005.HK (2001/05-2001/07)



(b) Stock 0005.HK (2001/08-2001/10)

Fig. 1. Trading Volume, News Articles and Historical Price Movements With Conflicts

presented in the market news articles as well as the trading volume? (2) How to formulate a model by combining N ($N \geq 3$) data sources (e.g., historical prices, current prices and news articles impacts, trading volume) together? In this paper, we develop a two-tier multi-kernel learning framework (MKL) using a service oriented approach, aiming at adaptively and progressively integrating multiple kernels towards the improvement on both performance and effectiveness of the prediction. In the top tier, for different data sources, we develop a suite of data cleaning and data transformation techniques to provide a source-specific data modeling and preparation, which transforms and normalizes each source specific dataset of input source into the MKL ready data stream representation. In the next tier, we first train the models with single information source to generate a source-specific sub-kernel. Then we perform iterative learning to adjust and tune the weights of sub-kernels, and finally we predict the market volatility based on the cross correlation analysis of MKL learning results. We evaluate our MKL method over the HKEx 2001 stock market price dataset, stock trading volume dataset, and news articles of year 2001. Our experimental results show that the service oriented development of our multi-kernel learning approach can seamlessly integrate multiple sources into our market volatility analysis framework and increase the predication accuracy significantly.

The rest of this paper is organized as follows. Section II gives an overview of our proposed two-tier system architecture and focus on the tasks to be performed and the techniques to be used in the top tier of the MKL system. Section III presents the main tasks in the second tier of our system. Experimental results and discussion are reported in Section

IV. The conclusion and future work are given in Section V.

II. Architecture Overview

The multiple data source prediction system has a two tier architecture. The top tier is dedicated to preparing the datasets from multiple information sources to make them ready for the predication tasks in the next tier. It is composed of two major parts. The first part is data preprocessing. The second part is the data alignment. Since the news release time is independent of stock trading, we should align the relevant news articles to the market trading data.

The second tier is dedicated to the market volatility analysis and prediction through the model integration and training, which uses multiple kernel learning methodology to train the classifier. It consists of three tasks: First, we build one sub-kernel per source. Second, we learn to update and tune the weights for sub-kernels through weight adjustment methods, and finally we perform the multi-kernel based cross-correlation analysis of market volatility.

In this paper, we will give an instantiation of the multi-kernel learning method in the context of three information sources: historical price, news articles and stock trading volume. In the rest of this section we will use these three information sources as an example to illustrate the key techniques involved in the data preprocessing and data alignment tier. We will describe the three tasks of tier two development in Section III.

A. Data Preprocessing

1) News Articles: News articles are regarded as raw materials that need to be preprocessed before aligning them with the time series of historical stock price data. Given a set of news articles, our objective is to extract some useful information in them. For example, we are interested in the information that will trigger the stock price changes. One of the key challenges in data preprocessing phase is to determine how to extract useful information from the raw data sources. Some researchers align the contents of the whole news articles to the stock prices movements. We believe that only certain terms from the raw news articles may have high likelihood to trigger the stock price changes. For text document based information sources such as news articles, we need to perform the following four steps in the data preprocessing phase:

- **Language-specific term segmentation and refinement**. This step is to segment each of the news articles into a collection of terms. Documents written in different languages will need different term segmentation techniques. For Chinese news article, we perform term segmentation by utilizing an existing Chinese segmentation software[1]. Although the segmentation software could produce outputs with high quality, many words that are specific to finance domain cannot be segmented correctly. A finance dictionary is thus employed to refine the segmentation results.
- **Term normalization and filtering**. This step performs two tasks: (1) remove stop word and (2) filter out other unimportant terms/words and keep representative terms, such as nouns, verbs and adjectives.
- **Feature selection**. For long articles, not all the terms/words will be included into the final feature list. For example, Feldman [2] (Chapter IV.3.1) selects about 10% of the words as features. After the filtering step, we get a word-frequency matrix with 7052 term/words. We compute χ^2 score of these words and select 1000 terms/words with highest scores as features of news.
- **Term weighting**. With the selected 1000 features, we calculate the widely used $tf \cdot idf$ value for each term/word selected as its weight.

2) Historical Prices: With the development of high frequency trading, stock tick data is popularly used for market volatility prediction. It is important to note that tick data are distinguished from daily data in the following aspects.

- **Big amount**. Tick data have much more records than daily data over the same time period.
- **Unordered**. Tick data is not recorded by the order of their time stamps, but by the time they arrive at the logging system.
- **Variant interval**. Time intervals between different tick records may not be the same. As transactions may happen in any seconds, a time interval between consecutive records is not always the same.
- **Incomplete or missing Items**. The tick data may also arrive with some incomplete information or missing information due to collection errors or transmission errors.

Thus, raw tick price data is preprocessed through the following two steps:

- **Sorting**. Since transactions do not arrive in the order of their time stamps, we must first sort the entire list of tick records by their time stamps. This allows us to perform time series based alignment at a later stage.
- **Interpolation**. Since time intervals between consecutive transactions are not the same, and sometimes there exists no record for some time periods, which leads to one problem: In order to perform time-aligned correlation analysis we need to determine what price value should be filled in during that time period. There are two common ways to address this problem: (1) linear time-weighted interpolation, and (2) nearest closing price. The former takes into account the price information in the neighboring periods and place higher weight to the nearest neighbor period to compute a price value during interpolation process. The later method splits tick data in a given time unit interval (e.g., a minute) and samples the closing price in each time unit (e.g., per minute). If there is no record in a given time unit, then the closing price of previous time unit will be taken as the closing price of this time unit. The choice of time unit depends on the time series alignment requirement and the tick data sampling rate. Both approaches have their pros and cons.

[1] http://ictclas.org/

In the experiments reported in this paper we used the second method, as it is simple and easy to implement.

3) Trading Volume: Trading volume as time series data shares many similarities in terms of data characteristics and thus the preprocessing steps. First, it is similar to extract the trading volume from the high frequency trading records as the process of extracting price data. Since the value of trading volume data fluctuates sharply in the market, in order to minimize the noise in the classifier training process, the trading volume ratio vr, the difference between the current trading volume and the previous trading volume normalized by the previous trading volume, can be used to represent the trading volume data series.

Thus, raw trading volume data is preprocessed also using the following two steps.

- **Sorting**. Given that the trading volume is extract from the transaction records, and the transaction records got from the data source are unordered, in order to prepare for the alignment of the trading volume with news and historical price, we first need to sort the trading records.
- **Interpolation**. There might have lots of transactions during a time period T_i. In this circumstance, we summarized the trading volume of each transaction during this time period as the trading volume v_i during time interval T_i. For those time periods which do not have any transaction record or its transaction records are not readable, we consider no trading activities occurred and label it as zero.

Upon the completion of source specific data preprocessing, we have cleaned and tidied every information source based on our multi-kernel learning framework. In order to construct a sub-kernel for each information source in terms of multi-kernel learning objectives, we need to perform data alignment to finalize the set of alignment operations that we need to perform before we enter the tier two learning phase.

B. Data Alignment

Support Vector Machines (SVMs) are popular for classifying time series data. In order to use a SVM classifier, we need to re-sample the time series to a common length or to extract fixed number of features before applying static kernels. We call this type of alignments the classifier-specific alignment. In addition, we also need to correlate one information source with another, such as correlating news articles with historical price time series, we refer to this type of alignments as correlation-based data alignment.

1) Alignment of Market Price Time Series: Given that our goal for market volatility analysis is to predict the upcoming trend of market price in terms of going up or down or on hold, this can be done using the rate of change for a given period n on a time series P, often referred to as relative difference in percentage (RDP):

$$RDP - k(P_k) = 100 \times \frac{P_t - P_{t-k}}{P_{t-k}}$$

k is the prediction horizon. Based on this preprocessing, we can examine many different input time series and their performance in conjunction with the output series and the best results typically are achieved by using a multidimensional input vector consisting of several rates of change with different periods. For example, we can incorporate the time series RDP-1, RDP-2, RDP-3, RDP-4, RDP-5 and so forth. As a result, the different values at each time represent by which ratio the current price differs from a distinct price in the past.

Another advantage of transforming the original market price time series into relative difference in percentage of price (RDP-k) is to preserve the sequential dependency of the price time series. If we naively take the 30 price points as 30 independent features, the sequential dependency of the price serial p_{-30}, p_{-29}, $\ldots$, p_{-1} is not preserved and thus the machine learning model will not be able to use the sequence information. Several existing work [1] have shown that this preprocessing will improve the predictive power.

From the trading perspective, the prediction horizon should be sufficiently long to avoid excessive transaction costs in over-trading resulting. But from the prediction perspective, the prediction horizon should be short enough since the persistence of financial time series is of limited duration. Moving average (MA) is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles, and MA-5 is mostly common used in the prediction over time series. Thus in the experiments reported in this paper, we set the value k as 5 and the calculations for the RDPs indicators are given in Table I.

TABLE I
THE FORMULAE OF RDPS

RDP	Formula
RDP-5	$100 * (p_i - p_{i-5})/p_{i-5}$
RDP-10	$100 * (p_i - p_{i-10})/p_{i-10}$
RDP-15	$100 * (p_i - p_{i-15})/p_{i-15}$
RDP-20	$100 * (p_i - p_{i-20})/p_{i-20}$
RDP-25	$100 * (p_i - p_{i-25})/p_{i-25}$
RDP-30	$100 * (p_i - p_{i-30})/p_{i-30}$

In addition to RDPs, one can also employ some other popular market indicators from stock technical analysis. Table II lists the formulae of some of such market indicators. where p_i is the price at minute i, q refers to the order counted in minute, RSI(q) is relative strength index, RSV(q) is raw stochastic value, R(q) is William index, BIAS(q) represents the bias of the stock price, and PSY(q) means psychological line.

TABLE II
OTHER MARKET INDICATORS

Indicator	Formula
RSI(q)	$100 * UpAvg/(UpAvg + DownAvg)$
	$UpAvg = \sum_{p_i \geq (\sum_i p_i)/q}(p_i - (\sum_i p_i)/q)$
	$DownAvg = \sum_{p_i < (\sum_i p_i)/q}(p_i - (\sum_i p_i)/q)$
RSV(q)	$100 * (p_0 - \min_q(p_i))/(\max_q(p_i) - \min_q(p_i))$
R(q)	$100 * (\max_q(p_i) - p_0)/(\max_q(p_i) - \min_q(p_i))$
BIAS(q)	$100 * (p_0 - (\sum_i p_i)/q)/((\sum_i p_i)/q)$
PSY(q)	$100 * (\sum 1\{p_i > p_{i-1}\})/q$

2) Alignment of Stock Trading Volume Time Series: The stock trading volume data can be extracted from the tick-by-tick trading records. In order to predict the volatility of the stock price movements, we can re-sample the trading volume time series V in terms of the rate of trading volume change, which is defined as the relative difference in percentage of trading volume, denoted by vRDP.

$$vRDP - k(V_i) = 100 \times \frac{V_i - V_{i-k}}{V_{i-k}}$$

In order to ensure that the features of trading volume data that we feed into the SVR classifier learning model are independent, we also convert the trading volume data into some trading volume indicators using the formulae of vRDPs before the training of model. Table III lists a set of vRDP formulas for the volume indicator conversion, which we use in the experiments reported in this paper.

TABLE III
THE FORMULAE OF VRDPS

vRDP	Formula
vRDP-5	$100 * (vr_i - v_{i-5})/vr_{i-5}$
vRDP-10	$100 * (vr_i - v_{i-10})/vr_{i-10}$
vRDP-15	$100 * (vr_i - v_{i-15})/vr_{i-15}$
vRDP-20	$100 * (vr_i - v_{i-20})/vr_{i-20}$
vRDP-25	$100 * (vr_i - v_{i-25})/vr_{i-25}$
vRDP-30	$100 * (vr_i - v_{i-30})/vr_{i-30}$

3) Alignment of News Articles with Market Price Time Series: This type of alignments is to prepare and align other information sources for market volatility analysis based on their correlation with the extracted historical price features. This enables us to train machine learning model with information from multiple sources.

In the context of news articles, we need to extract news features that can correlate with the stock price volatility. Given a set of stocks, $S = S_1, S_2, \ldots$, where a stock $S_i = s_{i1}, s_{i2}, \ldots, s_{iT}$ is a sequence of stock prices in the time interval T (for example at 1 minute sampling rate). In the same interval there exists a set of news articles $A = A_1, A_2, \ldots$, where a news article A_i contains a set of features (such as terms) f_{ij} ($j = 1, \ldots, m$). We assume that it is known which stock S_i an article is related to. We represent a feature f related to a stock S_i in the time interval T as a time series, $f(i) = f^i(1), f^i(2), \ldots, f^i(T)$, where $f^i(t)$ is defined as follows:

$$f^i(t) = \frac{AF_{i,f}(t)}{N_i(t)} \times log(\frac{N_i(A)}{AF_{i,f}(A)})$$

where $AF_{i,f}(t)$ denotes the number of related articles in A containing the feature f for the stock S_i at time t and $N_i(t)$ denotes the total number of articles in A which are related to stock S_i at time t. Here t is a time unit defined by user, such as one minute, one day, or one week. We call f^i the adjusted TF-IDF value of the news feature f related to stock S_i at time t. Similarly, $AF_{i,f}(A)$ denotes the number of articles in A that are related to stock S_i and containing the feature f during the interval T and $N_i(A)$ denotes the total number of articles in A which are related to stock S_i during the interval T.

Similarly, we can also align news with trading volume time series.

In principle, with a total of N information sources, if our goal is to predict the volatility of market prices, we need to align all $N - 1$ sources with the stock price time series.

In addition, one can also introduce different weights for different features. Due to space constraint, we omit this normalization step in this paper.

III. MULTI-KERNEL LEARNING AND CLASSIFICATION

In this section we describe the main functional components in the second tier of our MKL framework, including training and building kernels for different information sources and integrating multiple sub-kernels through a chosen machine learning model for classification based prediction.

A. Model Integration and Training with MKL

1) Model Integration: In order to do the model integration, we explore the idea of 'incremental learning' to construct five different prediction models for companion and better understanding of the effectiveness of our MKL model. Firstly, we use some single information source based models to do the prediction. According to the difference of the data sources, we named it as news article only model which only use news data for prediction, and historical prices only model which only use historical price data for prediction. We then make some naive combination about the data source, and propose a model named naive combination model which simply combines the news data and price data together. Thirdly, we explore the multiple data source which named as MKL model. In order to show the significance of incorporating trading volume in MKL prediction, we compare the performances of the model of using news data and price data only (MKL-NP) with the model of using news data, price data and trading volume data together (MKL-NPV).

- **News article only** (Model 1) . This model takes labeled news instances as the input of SVM. It tests the prediction ability when there are only news articles.
- **Historical prices only** (Model 2). This model takes labeled price data as the input of SVM. It tests the prediction ability when there are only history prices.
- **Naive combination of news article and historical prices** (Model 3). This approach uses the simple combination of news articles and prices. Naive combination means combine the 1000 features from news and 11 features from indicators to form a 1011 feature vector.
- **Using multiple kernel to combine news articles and historical prices** (MKL-NP) . Based on model 3, this model uses multiple kernel learning approach to analyze the news articles and prices data.
- **Using multiple kernel to combine news articles, historical prices and trading volume** (MKL-NPV) . Based on model MKL-NP, this model uses multiple kernel learning

approach to analyze the news articles, prices data and trading volume.

Many other prediction models could also be employed and integrated in this MKL framework. Due to the space constraint, we select a small set of representative models to be discussed and compared in this paper.

2) Model Training: Kernel based method such as support vector machines (SVM) has been successfully applied to classify the time series using characteristic attributes extracted from the time series and related datasets as inputs. Basically, SVM uses a hyperplane to separate two classes. For classification problems that cannot be linearly separated in the input space, SVM finds a solution using a nonlinear mapping from the original input space into a high-dimensional feature space, where an optimally separating hyperplane is searched. We call a hyperplane optimal if it has a maximal margin, namely the minimal distance from the separating hyperplane to the closed (mapped) data points (so-called support vectors). The transformation is usually realized by nonlinear kernel functions. There are many kinds of kernels can be used to transform the original feature space to a high dimensional kernel feature space, in which the SVM is used to find a linear classifier.

Suppose the training set $G = \{(x_i, y_i)|i = 1, \ldots, n\}$, where x_i belongs to some input space X and y_i is the desired target value for pattern x_i, and n is the total number of data patterns, SVMs based classifier is defined as:

$$\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\begin{cases} y_i\left(\langle \Phi(w), x_i\rangle + b\right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

where w is the vector of parameters defining the optimal decision hyperplane $\langle \Phi(w), x_i\rangle + b = 0$, and b represents the bias. $\frac{1}{2}\|w\|^2$ is the regularization item and controls the generalization capabilities of the classifier and can be tuned during the training process. The second term $C\sum_{i=1}^{n}\xi_i$ is the empirical risk (error). C is referred to as the regularized constant and it determines the tradeoff between the empirical risk and the regularization item. Increasing the value of C will result in the relative importance of empirical risk with respect to the regularization item to grow. Positive slack variables ξ_i allow to deal with the permitted errors.

The bottleneck for any kernel method is the definition of a kernel k that accurately reflects the similarity among data samples. However, not all metric distances are permitted. In fact, valid kernels are only those fulfilling the Mercer's Theorem and the most common ones are the following: (1) The linear kernel, $k(x_i, x_j) = \Phi(x_i) * \Phi(x_j)$. The value of the kernel equals to the inner product of two vectors x_i and x_j in the kernel space $\Phi(x_i)$ and $\Phi(x_j)$. (2) The polynomial kernel, $k(x_i, x_j) = (\langle x_i, x_j\rangle + 1)^d$,d is an integer. (3) The radial basis function (RBF), $k(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right), \sigma \in R$. It is also called the Gaussian kernel. The kernel parameter should be carefully chosen as it implicitly defines the structure of the high dimensional feature space $\Phi(x)$ and thus controls the complexity of the final solution.

In the experiments reported in this paper, we choose to use the radial basis function (RBF) as the kernel in single kernel based model. There are many parameters in the kernel function to tune. For RBF kernel function, there are two parameters γ and C to be optimized in the training progress. Detailed parameters tuning methods will be discussed in section IV.

For training model 1, model 2 and model 3, we use grid search and 5-fold cross validation to find the best combination of model parameters. As for MKL, parameter selection is a little bit different. As the best parameter combination for sub-kernels of news and indicators has been found during the training of model 1 and model 2, we just need to adopt the derived parameters and search the best parameters which are specific to MKL.

B. Multi-Kernel Based Classification

1) Multi-kernel Learning: For SVM, we need to test by cross validation to find the best parameters and find the best kernel. But for multiple data sources, it is too complicated to map the data with one kernel to highly nonlinear space. MKL uses multiple kernels to map the data to the other space and merge different kernels to try to get a better separation function F. In MKL, it combines many kernels and the weights of each kernel regulates its importance. The kernels can be different kernels, with different parameters or different data sets for the same labels. In the multi-kernel learning framework, the optimal kernel is learned from data by building a weighted linear combination of M base kernels. Each kernel in the mixture may account for different features or set of features. The use of multiple kernels can enhance the performance of the model and, more importantly, the interpretability of the results.

Suppose k_m (m =1,...M) are M positive definite kernels on the same input space X,

$$\min \tfrac{1}{2}\tfrac{1}{d_m}\|F_m\|^2_{H_m} + C\sum_{i=1}^{N}\xi_i$$
$$s.t.\forall i, m,\begin{cases} y_i\sum_{m=1}^{M}F_m(x_i) + y_i b \geq 1 - \xi_i \\ \xi_i \geq 0 \\ \sum_{m=1}^{M} d_m = 1, d_m \geq 0 \end{cases}$$

where d_m is the weights of subkernel m, and ξ represents the loose variables and C is used to control the generalization capabilities of the classifier and is chosen by cross validation. In our case, the learned (optimized) weights d_m will directly give a ranked relevance of each kind of data information we used in the prediction process.

2) MKL based Classification: In order to decrease the training time and increase the ability of prediction, MKL is applied in the multiple data sources prediction system. In the implementation of MKL, model selection was performed by building several kernels with different kernel parameters.

In our case, we use RBF kernel for SVM and multiple kernel learning (MKL). We employ multiple kernel learning to aggregate the information within news articles, historical prices and stock trading volume. And we apply a multiple kernel learning toolbox SHOGUN [5] in our experiment. Unlike naive combination model (model 3) which trains SVM using

$$f(\overrightarrow{x}) = \text{sign}(\sum_{i=1}^{m} \alpha_i l_i \mathbf{k}_{naive}(\overrightarrow{x}_i, \overrightarrow{x}) + b)$$

where $\overrightarrow{x}_i, i = 1, 2, \ldots, m$ are labeled training samples of 1011 features, and $l_i \in \{\pm 1\}$. In the MKL based models, similarity is measured among the instances of news, instances of price indicators and instances of trading volumes respectively. We construct three similarity matrices for each data sources through the Euclidean distance. And these three derived similarity matrices are taken as three sub-kernels of MKL and weight $d_{m,news}$, $d_{m,indicator}$ and $d_{m,volume}$ are learnt for sub-kernels,

$$\begin{aligned}\mathbf{k}(\overrightarrow{x}_i, \overrightarrow{x}_j) =& d_{m,news}\mathbf{k}_{news}(\overrightarrow{x}_i^{(1)}, \overrightarrow{x}_j^{(1)}) + \\ & d_{m,indicator}\mathbf{k}_{indiccator}(\overrightarrow{x}_i^{(2)}, \overrightarrow{x}_j^{(2)}) + \\ & d_{m,volume}\mathbf{k}_{volume}(\overrightarrow{x}_i^{(3)}, \overrightarrow{x}_j^{(3)})\end{aligned}$$

where $d_{m,news}, d_{m,indicator}, d_{m,volume} \geq 0$ and $d_{m,news} + d_{m,indicator} + d_{m,volume} = 1$, $\overrightarrow{x}^{(1)}$ are news instances of 1000 features, $\overrightarrow{x}^{(2)}$ are indicator instances of 11 features, and $\overrightarrow{x}^{(3)}$ are volume instances of 6 features.

For other types of information sources or sub-kernel combinations, similar distance based similarity matrices and kernel functions can be constructed and easily imported into our multi-kernel based learning framework.

IV. Experimental Results and Discussion

To validate the effectiveness of our service oriented MKL approach, we performed experiments on HK exchange datasets with three important market information sources: historical prices, transaction volumes and stock related news articles. Our experimental results show that the proposed multiple data source prediction system with MKL has better performance, higher degree of accuracy and lower degree of false prediction, compared to single kernel models. Also integrating both news and trading volume data with historical stock price information can significantly increase the prediction accuracy significantly.

All experiments reported in this paper are performed using a sever with 2.53 GHz CPU, 4G Bytes EMS memory and running on a Linux operating system. The average running time of our MKL algorithm is about 11 minutes with three types of data sources (historical price, news articles and stock trading volume), which is very close to the time (nearly 10 minutes) spent in the experiments performed on two types of data sources (historical price and news article). That means, the multiple data source prediction system has good scalability.

A. Data Sets

In order to evaluate the performance of the multiple data source prediction system, we used the following three kinds of data sources: 1) stock price data, 2) news articles and 3) stock trading volume.

- Market prices. The market prices contain all the stocks' prices of HKEx in year 2001.
- News articles. The news articles of year 2001 used in our experiment are bought from Caihua[2]. All the news articles are written in Traditional Chinese. Each piece of news is attached with a time stamp indicating when the news is released.
- Stock trading volume. The stock trading volumes are extracted from the corresponding stocks's trading record of HKEx in year 2001.

Time stamps of news articles and prices/trading volumes are tick based.

HKEx has thousands of stocks and not all the stocks are active in the market. We mainly focus on the constituents of Hang Seng Index[3] (HSI) which, according to the change log, includes 33 stocks in year 2001. However, the constituents of HSI changed twice in year 2001, which was on June 1st and July 31th. Due to the *tyranny of indexing*, price movement of newly added constituent is not rational and usually will be mispriced during the first few months. We only select the constituents that had been constituents through the whole year. Thus, the number of stocks left becomes 23. The first 10-month data is used as the training/cross-validation set and the last 2-month data is used as the testing set.

B. Parameter Selection

During model training period, parameters are determined by two methods: grid search and 5-fold cross validation. Take model 1's training for example, SVM parameters to be tuned are γ and C. For γ, algorithm searches from 0 to 10 with step size 0.2; For C, the step size is 1 and C searches from 1 to 20. Thus, there are totally $50 \times 20 = 1000$ combinations of parameters (in other words, 1000 loops). In each loop, 5-fold cross validation is executed to validate the system's performance, which equally splits the first 10-month data into 5 parts and use 4 of them to train the model and the left 1 part to validate. Among the 1000 combinations, the one with the best performance is preserved and used to configure the final model for testing.

For models 1, model 2 and model 3, we select the parameters in the same way. For multiple learning models (MKL-NP and MKL-NPV), we just adopt the γs which have already been selected in multiple models' training. MKL's parameter C is selected by the same method as the other models.

C. Experimental Results

The accuracy of the prediction is obtained by checking whether the direction of the predicted volatility is the same

[2] http://www.finet.hk/mainsite/index.htm
[3] http://www.hsi.com.hk/HSI-Net/

as the actual trend. For instance, if the prediction for the upcoming trend is 'UP' and the upcoming trend is really rising, then we say that the prediction is correct; otherwise, if the prediction is 'DOWN', but the upcoming trend is rising, then we say that the prediction is wrong. Here we use the formula based on precision and recall to measure the accuracy of prediction, which has adopted by many previous works [4].

$$accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

where tp and tn refer to the number of true positives and true negatives respectively. fp and fn denote the number of false positives and false negatives.

Table IV lists the cross validation results and Table V lists the results of independent testing (numbers in bold font indicate the best results at the given time point and the second best results are underlined). From these two tables, we can see that the multiple learning models (MKL-NP and MKL-NPV) outperform the other three single models both in cross validation and independent testing, except for point 5m in cross validation and points 10m in testing. Although both naive combination approach (model 3) and multiple learning models make use of market news and prices, naive combination does not outperform single information source based model as expected. On the other hand, due to a different learning approach, MKL models balance the *predictability* of news, prices and trading volume (Market news, stock prices and trading volume have their own characteristics and the information hidden in either of them could be a complement to the other.). Comparing to cross validation results in TableIV, although the performance of MKL models decrease in independent testing, MKL-NP can achieve 3 second-best-results and MKL-NPV can achieve 4 best-results and 1 second-best-result, which also proves that MKL models outperform than the other models. From these tables, it can be clearly observed that MKL-NPV has better performance than MKL-NP in most circumstances, which means that, taking the trading volume information into account is much helpful to improve the performance of the prediction system.

TABLE IV
PREDICTION ACCURACY OF 5-FOLD CROSS VALIDATION (%)

Model	5m	10m	15m	20m	25m
Indicator	**60.24**	58.42	57.59	58.48	57.99
News	59.12	62.65	62.84	63.06	60.93
Naive combination	60.05	61.78	62.84	62.84	60.85
MKL-NP	60.20	**62.94**	63.68	64.23	61.14
MKL-NPV	59.43	62.3	**64.12**	**65.29**	**62.56**

TABLE V
PREDICTION ACCURACY OF INDEPENDENT TESTING (%)

Model	5m	10m	15m	20m	25m
Indicator	53.48	50.23	47.84	48.92	45.38
News	52.94	51.13	44.40	52.38	53.41
Naive combination	56.15	**61.54**	49.14	52.38	45.78
MKL-NP	56.68	57.84	53.20	53.87	50.34
MKL-NPV	**59.1**	59.58	**58.25**	**56.69**	**54.39**

V. CONCLUSION

We have presented a service-oriented approach to market volatility analysis and prediction based on a multi-kernel learning framework (MKL). By service oriented in the context of market prediction, we mean that the users of our MKL service framework can feed the service with n sources of stock market time series datasets, and obtain the prediction of up, down or hold of a given stock in the forecasting time interval. The MKL service framework is designed using the two-tier service oriented architecture. The first tier is focused on data cleaning and data transformation techniques, which perform data alignment and normalize a source specific input dataset into the MKL ready data representation. The second tier is dedicated to developing model integration and normalization to prepare the datasets from multiple information sources to build one sub-kernel per information source and learning and tuning the weights for combining sub-kernels and perform multi-kernel based cross correlation analysis over market volatility. To the best of our knowledge, this is the first work that presents a service oriented approach to facilitating the incorporation of multiple factors (more than two) in stock market volatility prediction. We conduct experiments by using one whole year of Hong Kong stock market tick data. Results have shown that, 1) Multiple kernel learning methods have a higher accuracy rate and a lower false prediction rate compared to single kernel methods; and 2) Both news and trading volume information are critical sources of information for improving the effectiveness of stock market volatility predication.

ACKNOWLEDGEMENT

The first author's research is partially supported by the National Nature Science Foundation of China under grant No. 61103125, the Doctoral Fund of Ministry of Education of China under grant No. 20100141120046, the Natural Science Foundation of Hubei Province of China (Grant No. 2010CDB08504), and Research Project of Shanghai Key Laboratory of Intelligent Information Processing (Grant No. IIPL-2010-007). The second author's research is partially sponsored by grants from USA NSF CISE NetSE program and CyberTrust program, an IBM faculty award and a grant from Intel ISTC on Cloud Computing.

REFERENCES

[1] L. Cao and F. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks,*, 14(6):1506–1518, 2003.

[2] R. Feldman and J. Sanger. *The text mining handbook*. 2007.

[3] X. Li, C. Wang, J. Dong, F. Wang, X. Deng, and S. Zhu. Improving stock market prediction by integrating both market news and stock prices. In *DEXA (2)*, pages 279–293, 2011.

[4] R. Schumaker and H. Chen. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.

[5] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. Bona, A. Binder, C. Gehl, and V. Franc. The SHOGUN machine learning toolbox. *The Journal of Machine Learning Research*, 99:1799–1802, 2010.